REMARKS

Claims 1-29 are pending in the application.

Claims 1, 5, 14, 19, 20, 24, and 25 are amended.

Claims 10, 13, 15, and 16 are canceled.

No new claims are added.

Accordingly, claims 1-9, 11, 12, 14, and 17-29 are pending upon entry of this amendment.

The Examiner has rejected claims 1-29 under the obviousness provisions of 35 U.S.C. §103(a) as allegedly being unpatentable over U.S. Patent Publication No. 2002/0133412 to Oliver (hereafter referred to as "Oliver") in view of US Patent Publication No. 2002/0123993 to Chau et al. (hereafter referred to as "Chau"). Based on the above Amendment and the following Remarks, Applicants respectfully requests that the Examiner reconsider the rejection, and withdraw it.

As amended, independent claims 1, 19 and 24 recite:

> 1. A computer implemented method for dynamically rendering data in a markup language, the method comprising:
>
> identifying a symbol in the data in the markup language, the symbol indicating a query of a data set, the query containing one or more variables, each variable of one of a plurality of data types;
>
> augmenting the markup language to support the variables;
>
> accessing the data set in order to generate a resolution to the query;
>
> substituting the resolution for the query; and
>
> dynamically rendering the resolution as a part of the markup language, according to at least one rule associated with the markup language wherein said symbol can be used to dynamically render multiple data sets.
>
> 19. A computer program product for dynamically rendering data in a markup language, the computer program product comprising:

program code for identifying a symbol in the data in the markup language, the symbol indicating a query of a data set, the query containing one or more variables, each variable of one of a plurality of data types;
program code for augmenting the markup language to support the variables;
program code for accessing the data set in order to generate a resolution to the query;
program code for substituting the resolution for the query;
program code for dynamically rendering the resolution as a part of the markup language, according to at least one rule associated with the markup language wherein said symbol can be used to dynamically render multiple data sets; and
a computer readable medium on which the program codes are stored.

24. A computer system for dynamically rendering data in a markup language, the computer system comprising:
an identification module, for identifying a symbol in the data in the markup language, the symbol indicating a query of a data set, the query containing one or more variables, each variable of one of a plurality of data types, the identification module further adapted to augment the markup language to support the variables;
a data access module, for accessing the data set in order to generate a resolution to the query, the data access module being coupled to the identification module, the data access module further adapted to substitute the resolution for the query; and
a rendering module, for dynamically rendering the resolution as a part of the markup language, according to at least one rule associated with the markup language wherein said symbol can be used to dynamically render multiple data sets, the rendering module being coupled to the data access module.

The claimed method, computer program product, and system, as amended, enable data to be dynamically rendered in a markup language by identifying a query that contains one or more variables, augmenting the markup language to support the variables, generating a resolution to the query, substituting the resolution for the query, and dynamically rendering the resolution. The use of variables in query expressions is beneficial for several reasons. First, variables provide a notational convenience such that values are assigned to variable names. Second, the variables can have different data types such as numeric, node set, or Boolean besides a string data type. Third, different values may be assigned to the same

variable name, for example variables can be made to point to new documents by simply changing the document, within the variable value, to which the variable is bound. For example, rather than hard-coding a candle catalog document by referencing it as "file:///CandleCatalog.xml/Items/Item/No" within the markup language, the claimed query expression with variables allows the use of references such as "file:///catalog/Items/Item/No" where the variable "catalog" may have a value "CandleCatalog.xml" to point to a candle catalog document or "ComputerCatalog.xml" to point to a computer catalog document that conforms to the same schema or document type as the candle catalog document.

The combination of Oliver and Chau do not disclose the claimed method, computer program product, and system. Oliver describes a billing procedure that each service provider will use to charge the users of other providers for its service. The system also provides client authenticity verification by Token Validation Service (TVS). Oliver discloses a query-string presentation technique to transfer tokens where the query-strings that include the tokens are merely appended to the URLs of HTML pages generated and returned to the web browser ([0307], [0346], and [0373]). Oliver's query-string is merely appended to the URL following the syntax "?TVS=". This is not the claimed variable for several reasons. First, Oliver's query string is not present within a markup language and instead it is in the URL. Second, Oliver does not augment the markup language to support the query-string. Third, Oliver does not substitute a resolution for the query in the markup language. Lastly, Oliver does not render a resolution present as a part of the markup language. In addition, Oliver teaches away from the use of the query-string technique by disclosing difficulties of the technique ([0307]) and further stating that the query-string method is not ideal from a security standpoint ([0373]).

Chau discloses XML document processing consisting of creating metadata for searching the documents, generating documents from a SQL query or a relational database, and decomposing XML documents for storage in a relational database. In the office action, the Examiner does not even contend that Chau overcomes the above deficiencies and, in fact, Chau does not teach or suggest the combination of identifying in the markup language a symbol in the data that indicates a query of a data set with the query containing one or more variables, augmenting the markup language to support the variables, accessing the data set in order to generate a resolution to the query, substituting the resolution for the query, and dynamically rendering the resolution as a part of the markup language.

Accordingly, neither Oliver nor Chau nor their combination teach or suggest dynamically rendering in a markup language by identifying a query that contains one or more variables, augmenting the markup language to support the variables, generating a resolution to the query, substituting the resolution for the query, and dynamically rendering the resolution. Accordingly, Oliver and Chau do not disclose the claimed invention.

Accordingly, Applicants request that the Examiner reconsider and withdraw the rejection of claims 1, 19 and 24 along with all claims dependent thereon.

Conclusion

Applicants believe that all of the stated grounds of objection and rejection set forth by the Examiner in the Office Action have been properly accommodated or addressed. Applicants, therefore, respectfully request that the Examiner reconsider all presently outstanding objections and rejections and withdraw them. The Examiner is invited to telephone the undersigned representative if it is felt that an interview might be useful for any reason.

Respectfully submitted

Date: August 21, 2007

By: /John T. McNelis/

John T. McNelis
Attorney for Applicants
Reg. No. 37,186
FENWICK & WEST LLP
Silicon Valley Center
801 California Street
Mountain View, CA 94306
(650) 335-7133
jmcnelis@fenwick.com